

## GIS FUNCTIONS<sup>1</sup>

Ferenc SÁRKÖZY

Department of Surveying  
Technical University Budapest  
H-1521 Budapest, Hungary  
Phone: (36 1) 463 3212 Fax: (36 1) 463 3209  
E-mail: sarkozy@altgeod.agt.bme.hu

Received: April 15, 1998

### Abstract

The review of GIS development trends shows that the germs of two main directions can be observed even nowadays. On the one hand, the simplification of systems is urged by growing number of applications first of all related to the enterprise wide information systems, on the other hand, more complex modelling tools are required by different branches dealing with spatial decision making.

In both fields the standardization of data models, functions and that of processing control plays a decisive role. The introduction summarizes the reasons requiring the research of GIS functionality.

In the first section the paper discusses the possibilities of GIS function standardization with an emphasis on the processing environment.

The second section deals with the influence of the Open GIS concept on the standardization of functions.

The third section is devoted to the problems of user-friendly controlling interfaces.

The conclusion explains some ideas about the real ways of GIS development.

*Keywords:* GIS, GIS functionality, GIS tools, spatial data models, Open GIS, user interface.

### Introduction

Reviewing the development trends in GIS approaching the end of the century (SÁRKÖZY, 1996) I have found that in the near future the idea of a monolithic, general, universal GIS will be changed by two new types. The first will be a simple GIS with truncated functionality incorporated into the enterprise or office information system, the other should have tuned up functionality for solving the problems of geographical, environmental modelling, decision making, planning.

One can suppose that the first type will be sold in large amount for different platforms and networks and therefore it needs enhancements related interoperability with different kinds of office program packages working under a few operational systems in fashion.

For the second type it is probable that a wide ranging development will take place. One essential part of this development should deal with the problems of optimisation.

---

<sup>1</sup>Research partially supported by the Hungarian Science Foundation Grant No. T 016487

This topic is worth to be researched and one of my following papers I am going to devote to this problem.

However, the unified GIS functions can play even more significant role in the ‘simple’ GIS architecture, and due to the enormous number of possible users, this question should be solved first.

The first problem is to determine the scope of essential and supplementary GIS functions, to formulate their definition, to find input – output and environmental parameters and their defaults and to set up the groups of the functions. By solving this problem we should not forget the goal: to create user-friendly interfaces for technically easy control of the GIS procedures.

In the field of GIS we can find different types of standardization efforts. First, several data transfer standards were developed. Nowadays the standardization of geographic features is on the agenda. We can read also papers on the data models, however, it seems to me that this question should be solved together with the Open GIS effort. The most promising standardization activities are developing the open system concept, therefore in the section two I try to sum up its most important statements, and analyze their impact in relation to my topic.

The goal of the research is to create conditions for easy to use GIS software in different branches of administration, management, engineering, business, etc. and therefore it is reasonable to express some ideas about the user interfaces. This will be done in the section three.

On the basis of this paper and also the research published in (SÁRKÖZY, 1996) I will make as conclusions some comments in relation of the generally applicable GIS concept.

## **1. Types and Properties of GIS Functions**

The GIS textbooks use very similar listings and grouping for the GIS functions. The following set is based on Maguire and Dangermond, ‘The Functionality of GIS,’ in (MAGUIRE et al., 1991).

### **1. Capture**

- digitizing
- automatic-scanning, semi-automatic scanning
- Global Positioning System (GPS)
- Remote sensing
- stereo plotter
- OCR
- voice

### **2. Transfer**

- traditional media: tape, disk, CD-ROM

- network transfer, disadvantages and advantages: cheaper, faster, more efficient; maybe slow due to limited network bandwidth , overloaded system;
- problems with meta-data, etc.

### **3. Validate and edit**

- basic operations: add, delete, change
- errors associated with cartographic objects: point, line, polygon
- errors associated with attributes
- other errors yet to define and model

### **4. Store and structure**

- tessellation: regular (bitmaps, grids, RLE, Morton ordering, hierarchical tree structures), irregular (Triangulated Irregular Network)
- vector: unstructured (spaghetti, primitive instancing, entity-by-entity), topological (TIGER, DIME)
- hybrid: vaster
- data structures for attribute data: flat file, inverted list, hierarchical, network, relational

### **5. Restructure**

- a commonly performed operation could be very time consuming (e.g. tiger file to arc/info format)

### **6. Generalize**

- major operations: smoothing, simplification, aggregation, dissolving
- interpolation

### **7. Transform**

- affine: scale, rotation, translation, inversion
- curvilinear: map projections
- attribute: linear, non-linear

### **8. Query**

- spatial search
- overlay, spatial join

### **9. Analyze**

- overlay
- map algebra
- network modeling

### **10. Present**

- maps, graphs, statistical summaries, reports, tables, lists, etc.
- static vs. dynamic
- multimedia
- about effective graphic communication

### 1.1. Data Capture

Without studying deeply the list above we can immediately remark that the first point, the **data capture**, is not represented consistently. I have in mind first, that there is no hint at the attributive data capture, if not the OCR and voice are mentioned for it, secondly nowadays the capture of digital geographic data should be considered as the transfer of existing data or as an order for the mapping industry to produce the required data if they do not exist. Consequently, the node of data capture has two subnodes, the spatial data capture and the attributive data capture, and both are integrated only conditionally in the GIS if the organization maintaining the GIS has *complementary mandate* for the data capture, too.

The detailed analysis of the data capture node is out of the scope of this paper. However, some comments seem to be useful for the further discussion.

1. In the first period of GIS the applications required only low resolution geographical data, the digitizing of small scale paper maps in most of the cases could satisfy the tasks in question. Recently the spatial informatics is present in all categories of resolution and accuracy, the same software can be used for the cadastre, utilities, administration or environmental projects. The GIS user in general has nothing to do with surveying and mapping accuracy standards. He or she should know only what standardised resolution or scale of the input digital data is required by his or her particular task. Including the digitizing in the set of regular GIS functions might result in serious errors when induced by the easy instead of looking for the proper data the tasksolvers try to use data got by digitizing maps or plans of dubious origin.
2. We face with a completely other situation looking at the attributive data capture. Technically the attributive data, if those are not in digital form, should be typed in using SQL functions of the GIS software. Possibly this process can be made easier by the OCR technique. The essential part of the attributive data capture is usually in tight contact with the profession and institution of the GIS task solver. Due to the infinitely large number of possible attributive data the standardization in this field is significantly slower and therefore the particular institutions construct usually only in house data descriptions. That means that the responsibility for the adequateness, accuracy and completeness of these data lies on the institutions running the applications.
3. As mentioned above, filling up the database continuously can be realized by the SQL functions and therefore in the attributive data capture subnode of the first group remains only the OCR and the voice. We can consider these new techniques as parts of the supplementary GIS function group 'data capture'.

## 1.2. Data Transfer and Storage

In my point of view this group is the most important for the future GIS. According to my concept, the future GIS uses imported digital spatial data and in many cases the attributive data are also stored in databases of the institution or some partner organisations.

The main functions of the node are as follows:

- **read and store** the data in original format;
- **recognize and validate** the data;
- **transform the format** of the data;
- **transform to common reference**;
- **store the geographic** data using co-ordinates as keys;
- **georeferencing** the attributive data;
- **store the attributive** data.

The **read and store** function requires the possibility of the GIS to have networking modules and a built in function to store the raw data in an assigned system directory. The system should not make distinction upon the suffixes, the entry is allowed and the raw storage is classified on the basis of the metadata evaluation.

This evaluation is made by the **recognize and validate** functions. These functions work automatically and need unified metadata information in the header of every imported file. For the validation every opened project should have a set of keys corresponding to its resolution, accuracy and authentication requirements. The validate function compares the metafile data with the keys and opens the way for the forthcoming procedures. Depending on the project's keys the validate function can be activated later, too, prior to particular operations. The recognize function evaluating the meta data in the header works out suffices for the format transformation.

The **transform** function using the results of the recognize transforms the geographic data into the format of the particular system. At this point we should make some brief comments referring to (SÁRKÖZY, 1996) for more details. The theoretic capability of the transform depends on the data models of both the transferred data and the receiving system. If the import data are higher level of complexity as the system itself the transformation should reduce all useful information into the lower level's tools. For example, if we have exported a comprehensive object like a block consisting of houses as instances but our GIS uses the single object conceptual model then the transform should complement all instances (houses) with a table value (or empty link) relating to the block and also with another one relating to the house.

In the opposite case we should build higher level relations manually.

Usually the import data sets are either geographic or attributive, but especially in the future, linked together geographic and attributive data sets can more frequently occur. In the latter case the transform constructs all the information related to geometry, tables and links of the imported data set into the frames of the particular system.

Importing only attributive data the transform evaluates the record structure, automatically creates and fills up tables and reports about the characteristics of the new table.

The **store** functions are activated on this stage in the case of geographic and compound data sets. The single attributive data will be stored provisionally in a temporary storage place.

The store maps the co-ordinates on the storage medium facilitating the fast spatial queries.

If the store function plays the role we explained above, that is the imported data are transformed in the format of the receiving system corresponding to their model of implementation the processing relays to the next function group: the georeferencing. However, in the case when the imported data are of lower level complexity than the system's data model itself, an interim stage should be included: the possible automatic complexity lift, or as we commonly use to say it, the building of topology. It is obvious that this stage is really executable only in two cases: by importing spaghetti like vector structures and by importing simple raster data. In the first case we have in mind the well-known commands build and clean of the Arc/Info, in the second the region creating possibilities of the modern raster oriented software. Notice that the building of comprehensive objects does not belong to this function, because it cannot be performed automatically in the lack of supplementary information.

Maybe the **georeferencing** was and remains the most problematic group of functions from point of view of automation, that is of creating easy to use systems. The essence of the problem is in the different way the geographical and attributive data are produced. There were different attempts to solve the georeferencing, however, the general solution has not yet come. The solved cases (geocode in the cadaster, address matching using dime and tiger files, postcodes, etc.) are not general but linked to a particular field or data set. More general solutions can result only the joint development of standards for geographic and attributive data capture. Therefore even in the future GIS one should have manual tools for linking the attributes to the spatial objects. The interface explained in the section 3 has important role in facilitating this very complex task.

The circumstances sketched out above demand great flexibility, complexity and efficiency from the function group in question.

The main task of georeferencing is the linking of attributive data to geographic ones. But in some, most favourable cases this cannot be done if the data are not converted to a common local reference system or projection. Thus, we think that the **transformations between projections** belong to this group, too. This is the function, that can embitter the work of the GIS task solver not familiar with cartography or geodesy.

However, the transformation is easy to automatize if the data in their metadata header contain transformation parameters to a generally accepted local reference corresponding to the site the data describe. The term 'commonly accepted' is trivial in developed countries: it is the official co-ordinate system of the country. In other regions it can be one of the used reference systems. Following our recommendation

the GIS software does not contain transformation parameters, those are included in the data sets. However, in less developed regions, in the lack of official co-ordinate system the data should contain different sets of parameters and the user should choose the appropriate common one.

### 1.3. *Edit (Transform) Generalize*

These functions belong essentially to the software systems of data capture or data production.

The extent of their possible use in the task solving depends on the validation code of the particular task.

The *merge*, *dissolve* editing functions play important role by the feature selection based on the generalized attributes. Some other editing functions as *rotate*, *pan*, *cut*, *copy*, *paste*, *erase*, *snap*, etc. are useful for developing **designed** objects.

The original goal of the latter functions was to correct the data captured by digitizing (scanning). According to our concept of data capture, if those are applied to the fundamental data set then only for the sake of updates. In these cases the functions should be controlled by the specialists (surveyors, photogrameters, cartographers) which have mandate for geographic data production and validation.

The *generalization* was and remains the most complex task of cartographers demanding talent and skills. The automation efforts on this field did not lead yet to a comprehensive solution. One cannot expect that a GIS task solver, e.g. an insurance expert could possess this art perfectly. Thus, the function of generalization has to be included into the set of supplementary GIS functions, used mostly in the process of data production.

I put the *transform* in brackets in this group, because the only application for it I can imagine is the design of new constructions.

### 1.4. *Querying the Data*

The basic function of the GIS with truncated functionality is the widest possibility of querying. The different combinations of attributive and spatial queries are well-known and do not require further explanation. I want only to point to the necessity of the proper reporting on map sketches and also in tabular form.

New possibilities in querying appear when the data are organized according to the object oriented approach and the GIS software is capable to manage this model. In this case we can query, classes, subclasses, that is complex objects and also methods encapsulated in those.

The complexity of querying grows rapidly if we involve as targets the derived data, too. This issue is closely connected to the interpolation and will be discussed there.

### 1.5. Functions Deriving New Data

Usually these functions are discussed under the title GIS analysis functions. However, in my opinion it is worth to make a little bit more clarity and orderliness in this very important question.

The conceptual data models (SÁRKÖZY et al., 1995) describe the way we perceive the phenomena of the 3D world. If we are observing the natural environment we can face with functions, that describe the value of a particular feature in dependence of the place and the time. The most frequently used function is describing the heights of the Earth's surface. In this particular case because of the slowness of changes the time parameter usually can be neglected.

Due to the technical and financial problems of data capture as well as applicability of the observed data, some phenomena (e.g. the heights) are measured with discrete function values, others are determined as instances belonging to qualitative classes (e.g. land cover).

In the first case, theoretically, based on the measured values we could compute some approximate forms (e.g. polynomials) (SÁRKÖZY, 1994) given with a number of coefficients. However, the practice prefers two other ways. The TIN approach stores the co-ordinates and height values of the sample points and a rule to connect them in a unique manner. The DEM method interpolates height values on a regular grid and stores besides the global characteristics of the grid only the heights in the grid points.

The model of qualitative classes is essentially identical with the simplest conceptual data model used also for modelling of artificial objects. This conceptual model is usually implemented by the layer structure using the planar enforcement.

To condense the information using the object concept we have two ways: summarizing the available attribute types belonging to the same site by multiple overlay operations, or setting up an object hierarchy by constructing complex objects.

In the first case the more attributes we have the smaller the objects are in the output, but the result is also a set of simple objects with homogeneous attributes. In the second case the complex object has some common attributes but its parts have their own attributes, too. In this structure the attributes are not homogeneous for the entire complex object and therefore we have to use another data model, the object oriented one.

Let us return to our main topic, to the analytical GIS functions as follows:

- **interpolation;**
- **deriving characteristics;**
- **agglomerating** the data;
- **condensing** the data;
- **interaction computations;**
- **export – import** of derived and evaluated data.

The **interpolation** plays very important role in all cases when the GIS deals with continuous or quasi continuous phenomena observed in discrete points. Almost



in all applications at least one layer of this type, the layer of elevations, is present. Even the simple query to the system about the spots of a given height requires a large amount of interpolating computations. Multiple interpolations are used by deriving such features as contour lines. Thus the systems have to possess the ability to choose the method and parameters adequate to the data and task to be solved. This automatic control can be supported by the metadata header containing semantic information about the phenomenon itself expressed by the data.

Usually in two dimensional GIS we deal with approximated scalar valued vector functions, that can be represented geometrically by surfaces (of course the opposite statement is also true: the real surface can be represented mathematically by scalar valued vector function). There are **derived quantities** that **characterize** the function. The GIS should have tools to compute the approximated values of these characteristics. Such characteristics play very important role in entirely different fields of technology, science, economy or administration. For example for the heights we can derive in each point the slope (minimal, maximal, in a given direction), aspect, curvature. From these data the military can set out passable tracks for different off-road vehicles, the water management can design the drainage, the geography can model the erosion, etc.

We can also derive *statistical characteristics* from the function values. Especially interesting results can yield the statistical comparison of two functions.

The data **agglomeration** is a function quite similar to the generalization. It is very difficult to use grided point data together with area type simple objects utilizing the traditional tools of a vector GIS. To cope with this problem we can give intervals of the function value in correspondence with the actual task. The spots where the values are inside a particular interval will form a set of area objects and a consistent layer system will be set up. The same is valid for the derived quantities as slopes, etc.

As mentioned above, the overlay function of the traditional vector GIS **condenses** the information linked to a particular piece of the Earth surface (joins the attributes of the overlaid objects). This function should be also confirmed in the future simplified GIS so long as the layer concept in data capture will exist. We should not mistake this type of overlay for the other one that uses reclassification and binary multiplication for site selection. If the overlay has performed the condensation, a spatial query can select the convenient area without any supplementary step.

The **interaction computations** should be paid more attention in the future GIS.

The phenomena existing on the same territory can cause different consequences in dependence of their mutual values. For example, using the slope and soil category values we can compute the coefficient of run off, moreover adding the values of the average precipitation and the aspects we can model the land covered with inland water. The interaction computations can be useful applied to features, too. For example computing the rate of the areas of the lot and the house designed on it and taking into consideration the zoning code the building authority can make decision about the building permission. Therefore the computational tools should

be incorporated not only in the raster based but also in the vector based systems.

More complex, discipline specific modelling tools (e.g. optimization procedures, erosion modeling, spread of pollution, etc.) however, should not be included in the simplified GIS software. In the case the application demands such tools, properly adjusted data interface to the particular module should be present. The comprehensive solution to this question will be given by the Open GIS concept discussed in the next section. The complete realization of this concept seems to drag on for a long time that needs as partial solution implementation of functions to create several quasi standardized **export – import** formats for the original, derived and evaluated data.

### *1.6. Functions for Data Presentation*

Although we are aiming at the simplification of the systems, this function group disregarding the multimedia should not be simplified in capabilities, but only in control. The multimedia representation, if needed, should be solved by external packages.

The main task of the GIS software is to determine proper (standardized) default values for lines, colours, filling patterns, symbols, text. The same is valid for the table output. The system sets these values in concordance with the task code, country code, output level code and output scale. The default displacement of symbols and text could be modified by the dragging technique. Automatic computations should be performed by the system, in the case of three dimensional visualization, for finding some optimal views (best spatial effect, less covered areas).

The task code pertaining presentation can be divided in three main groups. The presentation in the first group is regulated by state mapping standards. For this group no manual changes in the default values are available.

The second group contains typical map products with quasi standardized presentation. For this group the manual adjustment of the values could be performed only in consistent way, that is, e.g. the changing of one text type results in the corresponding change of all other ones.

The third group enables fully free settings. However, also here the system has to offer convenient and consistent values.

Summing up the explanation above, the **presentation** node will contain the following functions:

- **adjust;**
- **symbolize;**
- **plot** (graphics);
- **print** (tables);
- **3D view;**
- **complex 3D view.**

We have to make up for the explanation related to the last function, that is missing in the previous paragraphs. Nowadays, the well-known old drape function does not satisfy the users (mostly architects) especially in urban environment, they need the visualization of blocks representing houses and other buildings, trees, etc. To make it the software needs models (geometrical substitutes) of the objects and their measures. High resolution models of 3D objects can be produced by 3D GIS or CAD modules. However, our simple GIS should be able to import such models for just visualization purposes and have simple geometric models of low resolution (parallelepiped, cone, cylinder) on its own.

I hope that this short compilation of the functions of the simplified GIS in the future shows clearly enough that the simplification is valid basically for the users and not for the software producers.

## **2. The Open GIS Concept and its Possible Influence on the Widespread Use of GIS**

Beginning in 1993 an American consortium named OpenGIS has started to work out conditions for realizing its system, that 'is defined as transparent access to heterogeneous geodata and geoprocessing resources in a networked environment'. 'The goal of the OpenGIS Project is to develop a comprehensive open interface specification which enables developers to write software that provides these capabilities' (BUEHLER et al., 1996).

Let us look at the circumstances, that stimulated the appearance of the OGIS effort. First, the qualitative changes in networks' topology, capacity and services should be mentioned. Secondly, similar attempts in general information technology, such as the CORBA (Common Object Request Broker Architecture) of the OMG (Object Management Group) (LINTHICUM, 1997) or the OLE/COM of Microsoft. Third, the very large amount of new spatial data in the future, that will be produced due to the deregulation of high resolution r.s. satellites. Fourthly, the growing gap between the commercial GIS software products and the users' requirements. Last but not least, the reasonable realisation of the US. National Spatial Data Infrastructure project, that needs the OGIS project to be successfully accomplished.

The basic idea of the OGIS system is well illustrated on the *Fig. 1*, taken from the OGIS Guide (BUEHLER et al., 1996).

The OGIS system is an object oriented distributed software system, that can access and manage all types of distributed spatial data and enable interoperability between applications on the desktop. There are three nodes specifying the system: the Open Geodata Model (OGM), the set of services manipulating the OGM data, and the interpreter of "Geographic Information Communities" semantics. Because the system is based on the object oriented approach it is worth to have a short summary about its main features.

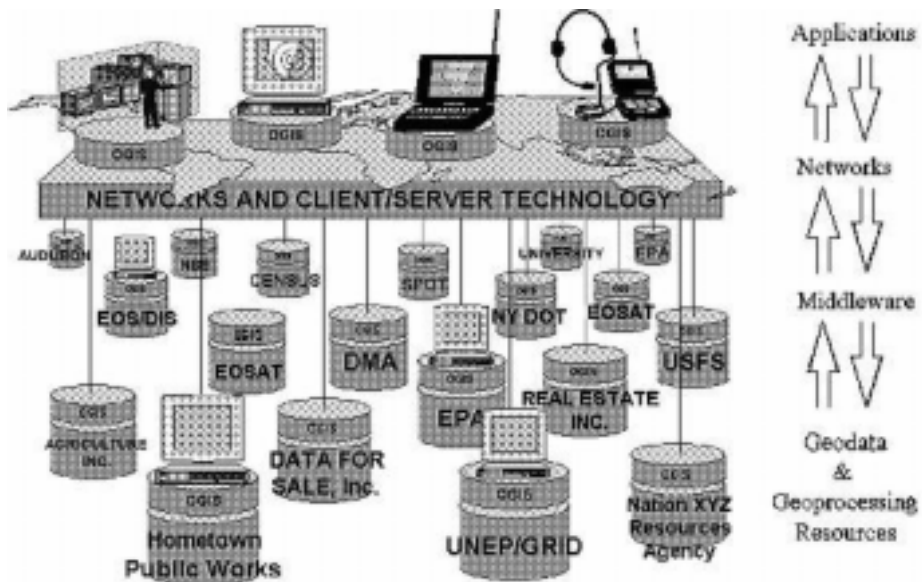


Fig. 1.

### 2.1. The Object Oriented Approach

Originally the object orientation was used as a programming concept leading to the construction of such programming languages as Simula, C++, Flavors, Smalltalk-80, Eiffel, etc. However, the object orientation developed further to create data base concept and data base management systems, e.g. the GemStone and the SIM.

Some decisive concepts are common in the programming and DBMS languages (PARSAEY, 1989).

The first concept is the *encapsulation*. It means that the object or some group of objects (class) and the procedures (methods) defined on it are stored and managed together. To activate a procedure the program sends a message to an encapsulated data-procedures set, in the consequence of the procedure's activity the set can send another message to another set, etc.

However, operations on spatial data can involve objects of more than one object class, i.e. the operation of intersection should be executable not only between two lines, but also between a circle and a line. To solve this problem the OGIS specification does not necessarily link the operation to the object, but makes possible

to choose an operation based on multiple object classes. This is done by introducing the concept of *types*. The type is a set of operations with a unique name. The types are organized in a acyclic hierarchy (from top to bottom, but not in the opposite direction), the union of two types is their common supertype, their intersection is their common subtype. There are different operations with the same name. When initiating an operation on the spatial data, the necessary operations will be selected from that type (subtype, supertype), which corresponds to the object classes of data involved in the operation.

The second concept is the *inheritance*. The inheritance is related to the class hierarchy. If we have a subdivision of a class, then the subclasses inherit from the class data and methods.

The third concept is the *object identity*. It means that despite different transformations the object's name should not change.

There is a fourth concept, the so-called *polymorphism* (AYBET, 1994). The word polymorphism we can interpret as different responses to the same message depending on the object in the address of the message. For example we can send a message: 'plot' to the addresses *a, b, c*. If into the object *a* is encapsulated a procedure of a circle, in the object *b* that of an ellipse and in the object *c* that of a square, then depending on the addresses, the command 'plot' will result in a circle, an ellipse or a square.

It is easy to conceive that using this model the application programs can be developed independently.

## 2.2. *The Relation of the OGIS Project to the General Information Technology*

The OGIS has a three level structure.

The first level is the *essential (abstract) model* of the real world's facts. This model comprises the most important conceptual data models supplemented with events (methods) associated to data.

The second level is the *specification model* which is the principal model of the OGIS software. It describes the content of the software and how the objects respond to the messages.

On the third level are the *implementation models* for different computing environments.

On the first and second levels the OGIS is independent from the real computing environment, the third level is realizing the link to different DCPs (distributed computing platforms) as COM, CORBA, DCE, etc. in frames of so called testbed projects. It is worth mentioning that although interoperation between different DCPs is out of the scope of the OGIS project, however, this goal is addressed by other IT activities.

At this point we can see a tight link between the OGIS implementations and the development of object technology and DCPs. Until the latter became mature enough the application programming interfaces (APIs) will play significant role in

many implementations.

The development of DCP leads to the change of the client server models.

At the first level of development the DCP manages the communication between the user interface and the underlying application. The application has internal interface to the data provider, which is bound to the database.

At the second level split from the applications emerge the application provider containing common applications and the data access provider with the possibility to communicate not only with regular databases, but also with collection of data files. In this model the DCP manages the data and process flow between the user interface, application, application provider, access provider, database or file group. Neither the application nor the application provider is in this case a monolithic GIS software, but the application is a well-suited part of the entire work flow, while the application provider possesses high level GIS functions.

The third level is the entirely object oriented model. In this model the applications are temporary associations of applets, and all kinds of services including access to different kinds of data are always available for the client.

The fact that the processing of spatial data is an organic part of the general information technology can be shown very well applying the *pluggable computing model*. In this model the computing is described as a mutual connection of the clients and service providers interfaced to each other. Using this model we can illustrate that the solution of spatial tasks is served not only by services specified by OGIS. Such not OGIS services are the data management, and the user interface.

The model defines the following services:

- Human-technology interface services (user interface);
- Tool services provide access to pluggable tools – additional functions which are plugged in the system (among others GIS functions);
- Data management services;
- DCP services;
- Operating system services;
- Hardware platforms.

The model defines also the interfaces (APIs) of each pluggable tool to the data management services, to the DCP services, to the human-technology interface services.

This model helps to understand the advantages of an open system in general and the OGIS in particular:

- the applications are *scalable* (for a simple map display we do not need all analytical GIS functions);
- the system is *extensible*, if new tasks emerge, new functions can be plugged in;
- the system is *diverse*, different types of applications and data types can be managed in the same system;
- the system supports *interoperability* that is it has integrated, co-operative and interchangeable tool sets.

### 2.3. The OGIS Data Model (OGM)

The OGIS tries to be very flexible and therefore allows interoperability and data access in all combinations, that is conventional to conventional, conventional to object oriented, object oriented to object oriented, etc. However, the mainstream is the dynamic transformation of geodata with different origin into a universal, single, object oriented data model, the OGM. The applications' programs act then on the data, transformed in this structure. The OGM can be interpreted as a *common language* without which the interoperability cannot be realized. It is based on a lexicon of common geodata types defined in terms of primitive data types available in all programming languages.

When we deal with data models we can distinguish conceptual model and implementation model (SÁRKÖZY, 1996). The implementation model itself can be also divided into data structure and data format.

The OGM conceptual model should be very general to be able to accommodate all types of existing geodata models, this is realized by incorporating into the model both the feature (object) based and phenomena based geographic data types.

The OGM contains three main components: the geometry, the attributive data and the metadata.

To fulfil the requirements of managing, processing and storing very different data types the OGM should have an object oriented database structure and should be constructed by object oriented programming tools. This approach allows associate proper procedures (methods) to each data type.

The vector and raster data types are called as feature and coverage, respectively. The geometry of the spatial data is defined for the 'Well-Known Types' (WKT). Their instances are called 'Well-Known Structures'. Using this constructs almost all spatial data types can be mapped into the OGM model. The descriptive part of the spatial data is linked to the related structure. The metadata characterizing the entire data set is associated with the top of the hierarchy. As mentioned in Section 2.1 the methods are not necessarily encapsulated in the objects. Such methods which are related to the entire data set, as for example reference system transformations, are encapsulated on the top level. Methods dealing with one object type (i.e. plot a circle) are encapsulated in the particular class, however, the methods dealing with different object types are chosen by multiple object classes.

### 2.4. Geographic Information Communities

A very important idea of the OGIS project deals with the different views of different specialities on the spatial data.

The general use of the existing spatial data is imaginable only if the different branches of users can understand and utilize each other's data. For finding the common denominator first we have to specify the independent communities, next we should divide the differing semantics into groups, finally we can choose one

of the following ways: provide pairs of geographic information communities with software to configure semantic translators (this is the OGIS approach) or standardise the semantic groups in a general way after what all communities can be translated to each other mediated by the standard.

Most of the intercommunication problems can be solved by proper standard of metadata. Among others the metadata should contain the accuracy of the features, the accuracy and resolution of the pixels, the date and completeness of the data (related to each included feature), the standard of the data set (if it is a standardized product, i.e. topographic map in scale 1:10 000), the reliability of attributes. Probably this latter is the less elaborated issue demanding further efforts in almost every information community.

In my opinion, however, in context of different branches there are problems that cannot be solved and should not be solved by translations, but by spatial standard enforcement. This is the preservation of the original accuracy of the implemented spatial features by archivation. For example, studying the hydrography on a digital topographic map we have right to omit the highway network, but have no right to store our product in a common access data store with sketched out by ourselves highways.

The geographic information communities can be divided into spatial data users and spatial data producers. The latter have the duty to determine the conditions for using, storing and disseminating spatial data.

### *2.5. The OGIS Service Model*

The OGIS operations are designed to be realized by the OGIS service model. This model consists of fundamental functions, which support to implement applications. The functions are software objects with the object oriented technology types.

The most important task of the system is to ensure access from the client applications to data in different structures in different databases.

The realization of data access works with a series of catalogs. The user defines a catalog entry schema which leads through catalogs predefined according to the existing data sets to the required features. Each selected catalog entry must have properties that match a part of the elements of the user defined schema. After the data are discovered and seem to be useful for the client he can select by a query the necessary part of the data mapped in OGS and read it into the application.

The access service maps the accessed data in OGM corresponding to their original structure and reference system. If the user needs the features in an other OGM structure (i.e. a vector object in raster format) or in an other OGM reference system (i.e. HD 72 instead of WGS 84) then he should invoke the transformation services. This can be made either in the geodata access process or directly from the application.

From point of view of our topic the query and geoprocessing services play a very significant role. The query service provides possibility to query the distributed



data stores using a standardized spatial query language. This language should have tools for spatial, attributive and semantic queries. The data stores depending on their development can process entirely or only partly the query. In the second case the other part of the query is solved by the application itself or by the geoprocessing service.

The geoprocessing service is a high level collection of GIS functions, that are not necessarily contained in an application and are accessible for all clients. The implementation of this tool kit might vary from the basic OGIS service to the distributed pluggable functions delivered by different vendors. This latter can also mean that the GIS application can access functions originally intended for not geographic application.

### *2.6. Prospective Influence of the OGIS Concept on the Widespread Use of Geographic Information*

As it was shown in (SÁRKÖZY, 1996) the future of GIS depends on its capability to become an integral part of the general information technology environment. To realize this, some conditions related to technology, usage and data access should be fulfilled.

From technological point of view the OGIS concept fits the geographic data access and geoprocessing in the distributed computing environment, that is it corresponds to the integration of GIS into the general information system.

The future GIS should be easy to use, that is based on controlling tools fitted in the overall computing work flow of the enterprise. The OGIS allows to construct proper user interfaces (Human-technology interface services) but does not specify those, simply leaves their specification to the general DCE. That is implementing the OGIS in a particular DCE the software developers should create controlling tools for this environment mirroring the general style of the complex application and in the same time expressing the fundamental functions of geoprocessing.

The strongest accent in OGIS is given on the access and transformation of existing geographic data. This gives a very significant support in establishing of the future GIS. However, the main obstacle to a general solution of the geographic data production is only partially discussed in the project. This problem is the lack of georeferencing (or in some cases not standardized georeferencing what is treated in connection with Geographic Information Communities) by a large amount of attributive data capture.

Summing up the benefits of the OGIS if widely implemented we can conclude that it will be a large step on the way to the future GIS, however, only one step.

### 3. User Oriented GIS Interface to Analytical Functions

As repeated several times in the present discussion, the future of GIS depends on the large number of new users, who are not, as a rule, GIS experts. These specialists of engineering, economy, administration, informatics, etc. will use the GIS only in the case, when they have nothing to do with different reference systems, projections, data validation, can easily perform, if needed, format conversions and can freely access geometric and descriptive data linked to each other and documented in a standardized way. Moreover, they need control tools to run geoprocessing functions that are conform with tools controlling other computing tasks and are easy to use.

To create a proper interface for GIS functions based on a monolithical GIS software, first the essential or fundamental GIS functions should be determined, next should be searched for those command combinations of the original software that correspond to the function in question.

The way the functions are invoked depends on the the computing environment. If the environment is graphical we can use self explaining icons to represent the function. We can in such environment also program a complex geoprocessing task using flow-chart.

Such interface system is the VGIS worked out in Vechta, Germany (ALBRECHT et al., 1996).

The VGIS system interfaces the GRASS GIS. The VGIS groups of fundamental GIS analytical functions are shown on the *Fig. 2*, taken from the referred article.

Although this is a reduced compression of the only analytical GIS tools related to the many thousand operations of a full fledged GIS, nevertheless, in my opinion, the analytical part of the fundamental GIS functions of the future GIS needs a further reduction.

Setting out from the VGIS analytical function collection after a new reduction we can reach a more compact set of fundamental analytical functions. For example, the tools for hydrologic modelling, as well as the functions of complex statistical analysis could be rightly excluded from a fundamental set. Moreover the distribution-neighborhood group, as it is shown by the authors itself, can be abandoned ordering its functions to the measurement (proximity) and to the search (cost/diffusion/search=reclassification) groups.

However, referencing the VGIS project we have not had the intention to analyze the correctness of its function groups but to show one way to create user friendly controlling tools for the GIS of the future. The VGIS is a shell over a monolithic GIS software, the future GIS will consist of a basic tool case supplemented if needed by distributed pluggable objects. Our future research aims at the construction of conceptual fundamental and supplementary function groups for the user interface in the OGIS environment.

The VGIS interface possesses also a graphical programming tool. The icons of the functions can be connected in a sequence and the resulted flow of control for the application will run as a macro automatically. If the process needs interactions

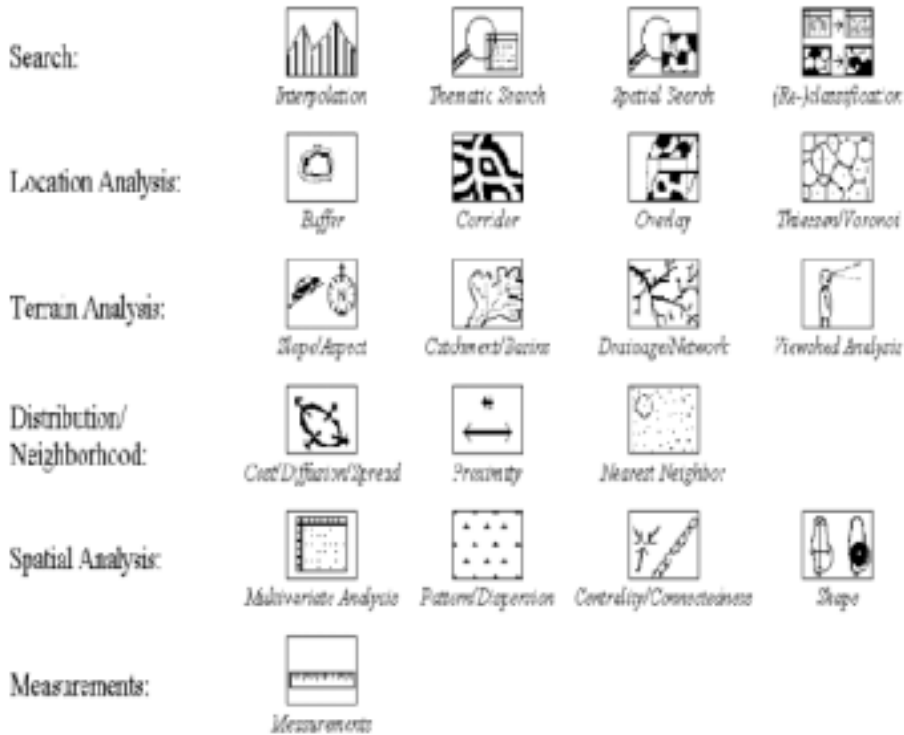


Fig. 2.

(data input, mode selection, etc.) during the execution, it stops opening a dialogue box for the required data. After the data entry the execution will be continued.

In the future the authors of the VGIS intend to apply their interface to the Arc/Info monolithic GIS software, but they also have the goal to fit the interface to the OGM data model.

#### 4. Conclusions

- The development of GIS as a powerful interdisciplinary tool for spatial reasoning, decision making, planning, engineering and inventory depends on the number of used systems. If this figure exceeds a particular limit the GIS will very fast become an organic part of the general information system wherever it would be implemented.
- The widespread use of GIS depends on several factors, we shall list only a few of those:
  - greater integration and accessibility of standardized digital spatial data;

- better database handling;
  - ability of flexible function expansion even with modules originally not designated for spatial data;
  - easy to use GIS with controlling tools incorporated in the overall work flow.
- The OGIS concept supports the realization of a good many of the GIS development conditions.
  - Independently from the platform, the environment, from the means and format of data transfer, the spatial data should have standardized header description, that provides the possibility of automated format, reference system and projection transformations and informs the user about the accuracy.
  - The masses of new users need unambiguous GIS function definitions and split of the function groups into fundamental and supplementary ones.
  - The OGIS concept and also the monolithic GIS software products should be supplemented with user friendly controlling tools like the VGIS interface.
  - The functions of visualization play equally significant role in all applications. Unfortunately the present discussion could not deal properly with this important topic. We hope in the future make up for this omission.

## References

- [1] ALBRECHT J., – BRÖSAMLE, H. – EHLERS, M. (1996): VGIS—a Graphical Front-End for User-Oriented GIS Operations. *International Archives of Photogrammetry and Remote Sensing*. Vol. XXXI, Part B2. Vienna 1996, pp. 78–88.
- [2] AYBET, J. (1994): The Object Oriented Approach: What Does it Mean to GIS Users? I., II. *GIS Europe*. Vol. 3., No. 3 and 4. pp. 38–41 and 46–47.
- [3] BUEHLER, K. – MCKEE, L., eds. (1996): *The OpenGIS Guide, Introduction to Interoperable Geoprocessing*. OGIS Project Technical Committee, Open GIS Consortium, 1996, Wayland (MA).
- [4] LINTHICUM, D. S. (1997): Reevaluating Distributed Objects. DBMS online, January 1997, Miller-Freeman, Inc.
- [5] MAGUIRE, D. J. – GOODCHILD, M. F. – RHIND, D. W., eds. (1991): *Geographical Information Systems: Principles and Applications*. London: Longman House, 1991, pp. 319–335.
- [6] PARSAYE, K. – CHIGNELL, M. – KHOSHAFIAN, S. – WONG, H. (1989): *Intelligent Databases*. John Wiley & Sons, Inc. New York, 1989.
- [7] SÁRKÓZY, F. (1994): The GIS Concept and the 3-Dimensional Modeling. *Computers, Environment and Urban Systems*. Vol. 18. No. 2, 1994. pp. 111–121.
- [8] SÁRKÓZY, F. – ZÁVOTI, J. (1995): Conceptual Data Model for Modeling of Scalar Fields and One Compression Method Usable for its Implementation. *Proceedings of the Fourth International Symposium of Liesmars titled: Toward Three Dimensional, Temporal and Dynamic Spatial Data Modeling and Analysis*. Liesmars, Wtasm, P. R. China, Oct. 25-27, 1995. pp. 1–10.
- [9] SÁRKÓZY, F. (1996): Prospects of GIS Approaching the 21 Century. *Periodica Polytechnica, Civil Engineering*, Vol. 40, No. 1, 1996, pp. 55–71.